Received: March 19, 2021. Revised: May 13, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

FDPHI: Fast Deep Packet Header Inspection for Data Traffic Classification and Management

Nahlah Abdulrahman Alkhalidi[1]* **Fouad A. Yaseen[2]***

[1]*College of Science, Computer Science Department, University of Baghdad, Iraq*
[2]*University of Baghdad, Computer Center, Iraq*
* Corresponding author's Email: nahla.alrahman@gmail.com

Abstract: Traffic classification is referred to as the task of categorizing traffic flows into application-aware classes such as chats, streaming, VoIP, etc. Most systems of network traffic identification are based on features. These features may be static signatures, port numbers, statistical characteristics, and so on. Current methods of data flow classification are effective, they still lack new inventive approaches to meet the needs of vital points such as real-time traffic classification, low power consumption,), Central Processing Unit (CPU) utilization, etc. Our novel Fast Deep Packet Header Inspection (FDPHI) traffic classification proposal employs 1 Dimension Convolution Neural Network (1D-CNN) to automatically learn more representational characteristics of traffic flow types; by considering only the position of the selected bits from the packet header. The proposal a learning approach based on deep packet inspection which integrates both feature extraction and classification phases into one system. The results show that the FDPHI works very well on the applications of feature learning. Also, it presents powerful adequate traffic classification results in terms of energy consumption (70% less power CPU utilization around 48% less), and processing time (310% for IPv4 and 595% for IPv6).

Keywords: Traffic classification, Packet header inspection, Neural network, Computer network.

1. Introduction

Internet traffic classification has become more influential with the rapid growth of current on-line applications and Internet networks. Accurate traffic classification has become one of the requirements for advanced network management tasks such as providing relevant Quality-of-Service (QoS), access control, fire-walling, and vulnerability assessment [1]. Traffic classification means the task where traffic flows are classified based on the type of service. Traffic flows are the group of packets that have the same source and destination port and IP addresses. The kind of service indicates to the application category of data flow within [2]. For instance, video conferences, multimedia streaming, and VoIP are examples of traffic flow classes. Multiple functions, including identification, tracking, optimization, and control, could be carried out on the traffic classes [3]. There are different network traffic classification techniques. We can categorize these techniques into four main categories they are port-based, payload-based inspection, statistical, and machine learning approaches. The port-based method is the earliest and the most simplistic one, which depends on extracting port numbers from the User Datagram Protocol (UDP) / Transmission Control Protocol (TCP) header fields of packets to conclude traffic classes [4].

The payload-based method commonly refers to as Deep Packet Inspection (DPI) technique, which depends on predefined models for various protocols to analyze the payloads [5]. Hence, authentic traffic classification needs to examine packet-payload. This procedure hardly is a choice due to (i) complexity constraints, (ii) packet-payload encryption, (iii) secrecy and constitutional matters [6]. The statistical

observations approach relies on the included information by a traffic flow, such as the packet size, the separation period between arrived packets, etc. However, this method does not utilize related characteristics and properties of a network that can provide a lot of important information, but it considers as the easiest way to classify the traffic flow passes through a network [7].

Machine Learning (ML) is an intelligent automaton that learns from activity utilizing data in its context and applies it to enhance the overall achievement based on designing effective and reliable prediction algorithms [8]. ML can broadly refer to as computational techniques applying expertise to develop performance or to obtain accurate expectation. The prepared data relates to the prior knowledge available to the trainer, which typically considers the form of analyzable data collected and delivered to the classifier of the neural network. This information could be modeled as labeled instruction sets based on the information gathered through interaction with surroundings according to the trained data. That means the characteristics and size of the trained data entering the neural network are essential to the success of the classification that the classifier performs [9]. Keeping in mind, applications trend for improving the encryption for more security and privacy. These applications utilized well-known security network protocols such as SSL, SSH, HTTPS, etc.

Current resolutions to application knowledge rely on ML and DPI. However, it provides a low precise at a real-time to classify the traffic of encrypted applications passes through the network [10]. So, traffic classifiers of the prospect will necessitate classifying traffic intelligently and efficiently depending on selected bits from packet headers instead of the whole field. That is to say, we decreased the computational burden by reducing check field bits by selecting bits from the packet header. Those bits represent the packet features, which are grouped as a set that can be executed with low computational complexity to promote the performance and to establish a traffic classifier that does not depend on encryption applications.

Our contribution aim to increase the performance of an ML that classifies the traffic flows, at the same time decreasing the computation time, power consumption, and CPU utilization by selecting specific bits to be checked from the packet header to decide the class of the transferred traffic flows through the network. The paper is organized as background and related work in section 2, while proposed system description will be in section 3, evaluation metrics are exhibited in section 4, performance evaluation are presented in section 5, and conclusion in section 6.

2. Background and related works

Earlier traffic classification studies have been extensively included in several different methods in this field, some of those techniques applied to ML approaches. We focused on works that used the ML in its researches to classify network traffic flows. According to classification levels, ML techniques can come into three kinds:

A. Supervised machine learning

The Supervised ML (SML) method depends on the provided pre-defined information. That means the input-output pairs are trained by a dataset, where the system executes the steps of a function that determines the output according to the input. This method needs to have a dataset describes the consideration of the system that can be employed to evaluate the achievement of this approach [11]. The Bayesian approach was applied by Moore et al. to distinguish protocols of the application layer. They got a higher accuracy by utilizing the variants refining. Several models were examined to improve the classification accuracy and computation performance, which involves C4.5, Decision Tree, Naive Bayes with discretization, Bayesian network, and Naive Bayes with kernel density estimation [4]. In [12], the Support Vector Machine (SVM) method was applied on three classes of predefined datasets, where a comparison of the average accuracy and performance between Bayesian and other approaches has been achieved. Finamore M. Mellia, M. Meo, and D. Rossi worked on the payload statistical as features inputs to SVM to classify network traffic flows [13]. In [14], the authors have applied traffic classification established on eleven well-known supervised machine learning algorithms, they verified 5 to 7 packets are the greatest numbers for the first step of the classification. Recently, many kinds of research went far inside packets even, it segmented the datagram into several bytes for performing traffic classification. These segments are input to recurrent NN to obtain the features which are represented by a vector of the entire datagram [15]. A. Duque-Torres, F. Amezquita-Su´arez, O. M. Caicedo Rendon, A. Ord´o˜nez, and W. Y. Campo provided a solution to tackle the issue of big-size traffic flows, which consume the network resources more than different flows consolidated. They proposed a method for investigating the usefulness of applying Knowledge Defined Networking (KDN) in big-size traffic flows classification by using ML [16].

B. Unsupervised machine learning

The most prominent feature of the Unsupervised Machine Learning (UsML) is that it does not rely on data preparation for building models that can distinguish patterns in a data flow. i.e., UsML can point associations in data traffic that the administrator may do not able to aware of or label it. That is, UsML employs unlabeled training datasets to generate patterns that are used by UsML to execute its job. A. Lakhina, M. Crovella, and C. Diot demonstrated that deals of packet characteristics (network and transport layers) recognized in traffic flow traces reveal the behavior and construction of a wide variety of anomalies. They showed that the investigation of characteristic distributions points to it allows profoundly sensitive detection of anomalies. Also, it permits the automatic classification of anomalies through UsML [17]. In [18], the authors analyzed the traffic based on three mechanisms utilizing structural and statistical models to automatically classify the traffic that employs the same protocol of the application layer. According to the flow content, they identified applications without requiring the port number. M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian introduced a single system that combines both classification phases and feature extraction. The system can manage both traffic features in which the data traffic is identified into principal classes and applications classification in which end-user demands [19]. The authors in [20] proposed a stacked auto-encoder to determine complicated connections of several sources of network data traffic through piling many essential Bayesian auto-encoders, which is trained on the objects utilizing the unsupervised knowledge policy. Besides, it is trained with the back-propagation algorithm that applies the supervised learning approach to capture the multiple relations across the network traffic flows. M. A. Lopez, D. M. Mattos, O. C. M. Duarte, and G. Pujolle presented a fast preprocessing approach for traffic classification that relies on feature normalization and feature correlation. They used a method that combines feature selection and normalization algorithms. Moreover, they evaluated the offered algorithms versus three diverse datasets for several ML classification algorithms. Their results showed a reduction in error rate and enhanced the accuracy [21].

C. Semi-supervised machine learning (SsML),

This method acquires its learn from both labeled and unlabeled datasets [22]. In general, SsML employs a large number of unlabeled data with a small number of a labeled dataset to prepare the ML to classify the traffic. In [23], the authors figured unknown applications as a particular classification dilemma with a lacking of the training dataset and processed it by offering a binary classifier that relies on a framework. Moreover, they proposed a method to obtain the unidentified information from a group of unlabeled data traffic, which is consolidated by a flow correlation and asymmetric bagging to ensure the pureness of evoked data. A. S. Iliyasu and H. Deng [24] utilized Deep Convolutional Generative Adversarial Networks (DCGAN) to develop a semi-supervised technique for traffic classification. Their work relied on small labeled datasets, but the achieved classification accuracy was lower than corresponding to other related researches. Based on the semi-supervised method, the fine-grained data traffic classification was proposed by Li G. Li, and X. Yu [25]. They used an algorithm constructs a matrix with many cluster centroids instead of single feature vectors. A small number of marked flows have been used by the algorithm to produce the supervised datasets. Afterward, the obtained unlabeled flows were merged with the earlier selected dataset to map an application. The semi-supervised technique is a significant research process for mining of data and ML because it utilizes a minimal of labeled features and picked unlabeled samples to prepare the system. In general, a semi-supervised learning approach can be used for semi-supervised classification and semi-supervised clustering purposes. Both techniques aim to increase the performance of learning based on limited labeled features [26, 27].

3. Proposed system description

Nowadays, the encryption of network traffic makes it very hard to get large labeled datasets that are needed by a supervised approach to training a deep ML model. On the other hand, the unsupervised method requires unlabeled datasets, but it suffers from the problem of low performance. Therefore, the best candidate ML approach is semi-supervised, due to it needs few labeled datasets to achieve the classification task. Moreover, to execute our proposal suitable for real-time classification, we employ flow statistical series features such as packet, byte, bit accounts, packet length, and packet direction, etc. In our proposed algorithm, we focused on the bit account to perform the traffic classification of a network, taking into consideration the packet and byte accounts. By taking the statistical of the packet, byte, and bits counts is that, isolated from being unique application identifiers, plus

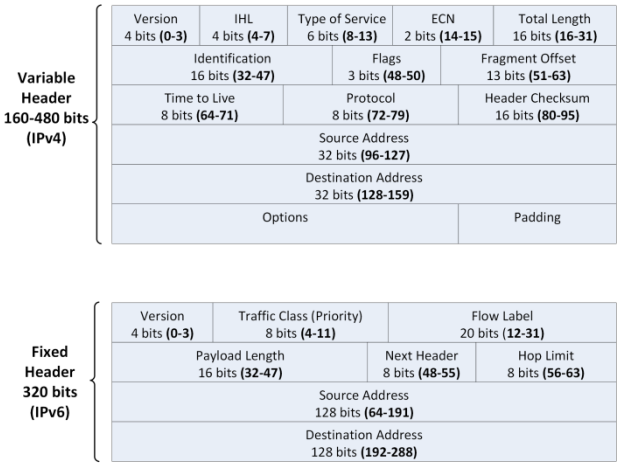

Figure. 1 Packet header fields

sequence- dependent of arrived packets. Moreover, the traffic subjects for checking can be processed as a flow (first byte as a guider) or as individual packets to be classified. Consequently, this leads to flexibility in selecting the labeled features and control the number of those features. Also, to reduce power consumption, processing time consumed, and increase the performance of the system. The position of the selected packet header bits refers to specific features that belong to the arrived packet to be classified by our model. For example, in simple words, we can classify the version of the IP address of the arrived packet by examining the third bit from the left of the version type field in the header instead of examining all the four bits in that field, as shown in Fig. 1.

A. Proposed system architecture

To classify and manage incoming traffic flow to a network according to the demands and states of the system (such as QoS, security, etc.), the network administrator can choose and determine features that affect to control and handle the data traffic through the support of the ML. The structure of the proposed system architecture, as shown in Fig. 2.

Traffic flow packets enter the system will subject to checking its header, which includes the features that can be used by the classifier to distinguish packets. The first 2 packets up to the first 20 have proven sufficient information to achieve acceptable classification precision even if the data traffic was encrypted [28]. The SsML approach based on a simple 1D - CNN is used to classify applications. The trained model predicts the statistical features of the whole flow from sampled packets with a large size of the unlabeled dataset. Later, the weights are transferred to a new model to re-train it to classify

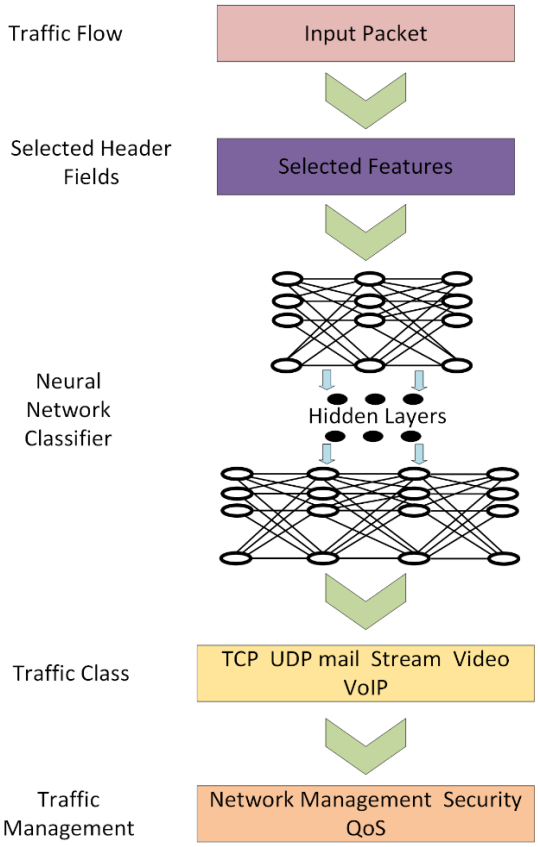

Figure. 2 System architecture

applications with simply some labeled samples.

B. Fast deep packet header inspection (FDPHI)

Within FDPHI technological innovation, we can examine packet header content of what crosses through Internet networks, treating it differently if needed. Thus, the proposed system can classify and manage traffic data based on the selected features (prioritizing some flow data, or banning the transmitted data, etc). DPI is a technique that can provide intelligence traffic flows classification on Internet networks. DPI can be applied to monitor, classify, protect data traffic, etc, to make decisions for transferring data on the Internet. Based on the header information enclosed in the packet, which carries all the description features of the data packet. The packet header bits can provide the packet features which can be used by the NN as input parameters. For instance, the packet header field (Version with 4 bits) value is 0100 for IPv4 and 0110 for IPv6. That means, examine only the effected bits that refer to a specific feature. While the portion (Protocol 8 bits) in IPv4, which is called Next Header in IPv6, values for the UDP 00010001 and the TCP 00000110. Packets serially enter the ports of communication devices, so the proposed system can be trained to check selected bits of the chosen field (or fields) from the header.

According to this concept, the proposed NN will execute and accomplish its duty to recognize and classify the traffic data flows. Also, this will reduce processing time, CPU utilization, power consumption, and increase the performance of the system. Our model approach employs the 1D Convolutional Neural Network (1D-CNN) depended on the assumption that sequential flow packets show correlated behavior. Implementing FDPHI model inspired by the method used in [29]. Therefore, we created a matrix P, which is produced by assembling a continuous series of packets' headers in a given flow. These packets' header is expressed as a vector H consists of the packet header bits. In IPv4, packet header bits are ranged between (160-480) and 320 for IPv6, as defined below:

$$H = [b_1, b_2, \ldots b_n], \quad n = \begin{cases} 160 - 480 & for\ IPv4 \\ 320 & for\ IPv6 \end{cases} \quad (1)$$

Where b_1, b_2, … b_n indicates features such as the number of packet header bits and its position in the header. In other words, the position of the selected bits specifies the chosen features. Hence, a single P_H is denoted as:

$$P_H = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ & \vdots & & \ddots & \vdots \\ b_{j1} & b_{j2} & b_{j3} & \cdots & b_{jn} \end{bmatrix} = [\, H_1, H_2, H_3, \ldots, H_j]^T \quad (2)$$

Where H_j is packet header features vector of the jth packet enters in the model. The dimension of each P_H is the number of features by the number of packets for a specific flow. The input to our model is the P_H, which is a 1D-dimensional vector with three columns cover (packet number, packet direction, and bit position). Packet direction is designated as 0 for the forward direction and 1 for the backward direction. Moreover, we normalized the bit position and packet number by dividing each by its maximum value. A schematic layout of the entire training method is represented in Fig. 3. We used 1D-CNN as a traffic classifier because it is suitable for sequential data [30].We applied sampled packets as an input to the proposed model, and as a data augmentation method, we sampled many times from various parts of the flow.

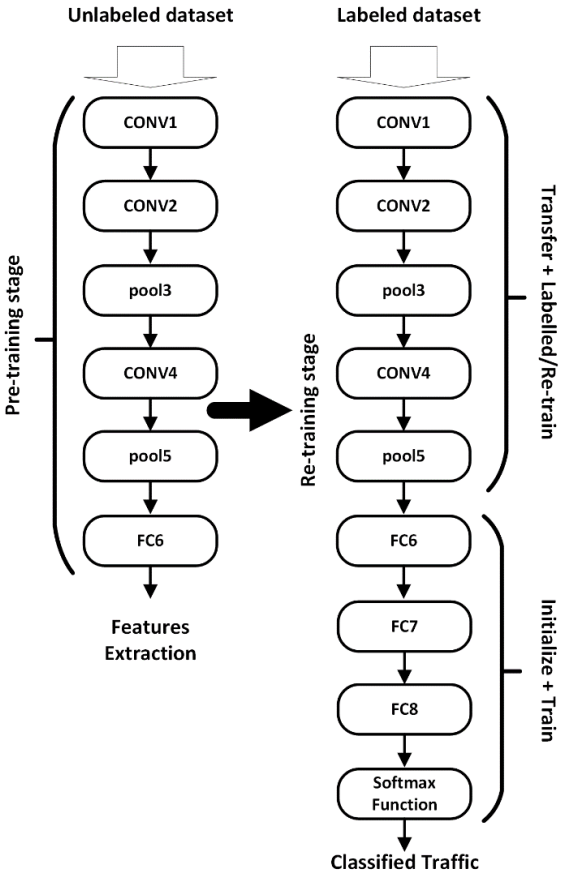

Figure. 3 Semi-supervised model

In the first stage, the unlabeled dataset is pre-trained via the CNN model. The resulting learned weights are transferred to the second stage of our model, which applied more linear layers. Then, the new model is used to re-train on small labeled datasets.

Table 1 represents the details of our proposed model structure. The employed activation function in our model is Rectified Linear Unit (ReLU) and Max Pooling. After Max Pooling and Convolutional layers, Batch Normalization is used to accelerate training.

C. Dataset collection

To build a characteristic dataset, we collected the live packet flows collected by our colleagues lab

Table 1. CNN model structure.

Layer	Conv1	Conv2	Pool3	Conv4	Pool5	FC6	FC7	FC8
No. of Filters / Neyrons	32	32	-	64	-	256	128	128
Kernal Size	5	5	3	3	3	-	-	-

Table 2. University of baghdad traffic statistics

Traffic Type	No. of Flows	% Flow	No. of Bytes (GB)	% Byte
P2P	560/320	8.31%	150.6	36.56%
HTTP	3,400,113	50.45%	205.3	49.84%
Email	604.781	8.97%	24.4	5.92%
Streaming	2,362.1	0.035%	8.2	2%
What's App	23,825	0.35%	1.5	0.36%
Data Base (DB)	2,105,738	31.25%	2.6	0.63%
Other	41,359	0.614%	19.3	4.68%
Total	6,738,498	100%	411.9	100%

affiliates at University of Baghdad (UOB). The communication services that have been used by the users involved different kinds of applications (Facebook, Skype, Whatsapp, Youtube, VoIP, etc). Table 2 lists the whole types of used applications and traffics that are included in creating the dataset. We captured the traffic pass through the network of listed traffic categories at the Internet access point (the gateway). Traffic aggregate represents all inter-networking sub-traffic of UOB staff and student activities for 2 hours every workday from May to December 2020. The recognized applications and their data sizes are summarized in Table 2. From Table 2 we can notice the HTTP and Database had a considerable part of the entire network flows 50.45% and 31.25%, respectively. However, Email contributed nearly 9%, while P2P recorded 8.31% of the total flow. This difference in the utilization ratio of network flows is due to management and administration policies. We impute this difference in the percentage usage of flows to the policies of the UOB campus network that is used by staff, faculty, and students. Also, non-academic content is strictly governed by the network policies of the UOB campus. Furthermore, the network infrastructure handles signature-based association to firmly throttle P2P flow. In contrast, the "Other" flows are used completely by students, are not actively controlled, and only apply manageable limitations of the bandwidth dedicated to each user. However, "Other" flows include many applications that have been recognized but are not related to a larger group and can be considered a frivolous part of the entire flows.

4. Evaluation metrics

To evaluate the performance of our SsML approach, we employed a 1D-CNN as a traffic classifier. We used three metrics to evaluate classifier performance:

- Accuracy: is used to estimate the classifier performance, which represents a ratio of correctly predicted of P_Hs, i.e., True-Positive (T_P) and True-Negative (T_N) to the overall recognized of P_Hs, which is the summation of T_P, T_N, False-Positive (F_P), and False-Negative (F_N). The following equation represents the accuracy,

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (3)$$

- Precision: is the ratio of accurately predicted positive f P Hs to the entirety predicted positive of P s, the precision can be expressed as,

$$Precision = \frac{T_P}{T_P + F_P} \quad (4)$$

- Recall: is the proportion of accurately predicted positive PHs to entire P s in the actual class, which can be given as, Recall: is the proportion of accurately predicted positive P Hs to entire Ps in the actual class, which can be given as,

$$Recall = \frac{T_P}{T_P + F_N} \quad (5)$$

To attune the hyper-parameters for our model, we divided 40 files for each specific class. Then, we trained the model with 25 labeled flows and confirmed with the other 15 flows. This trained data is harmonious with the postulated restriction number of labeled data that is transferred to the supervised training we managed. Our model showed an acceptable accuracy, and a deeper convolution did not provide higher accuracy. Moreover, fixing or retraining the convolutional portion of the transferred pattern through retraining did not significantly improve the accuracy.

Remark that we used the same hyper parameters for other tests without re-attuning them. Therefore, those sets of hyper-parameters appeared to be satisfactory over various datasets, which made our

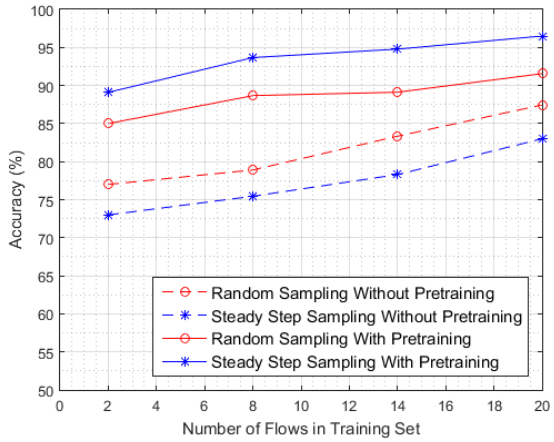

Figure. 4 Accuracy of supervised training set size

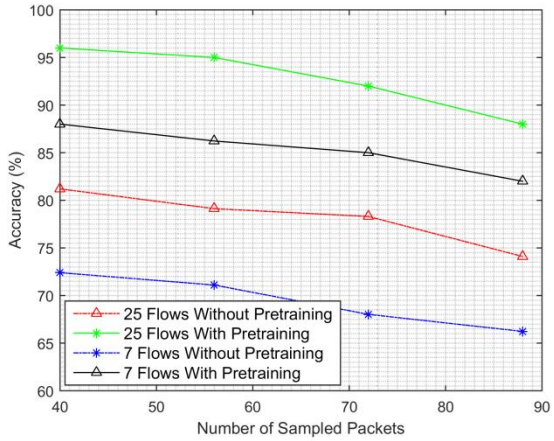

Figure. 5 Accuracy of steady step sampling

proposed method more efficient. Fig. 4 shows the accuracy of supervised training set for both random and steady step sampling methods.

Also, it shows that increasing the number of flows in the training set enhances the accuracy for a steady step more than the random sampling. Because random sampling starts from the flow head every 80 times we sampled a flow.

However, random sampling accuracy almost rises as the training set size increases. We think this is because forcing randomness through random sampling makes it harder for the proposed model to be suitable for real distribution. While in a steady step sampling approach, a flow is sampled by capturing different portions of the flow. Consequently, different patterns are provided by this method shows more data augmentation for those patterns. Therefore, the accuracy increases with increasing the training data set size. Also, Fig. 4 presents steady step sampling does better than random step sampling. Thus, the figure introduces the efficiency of our transfer

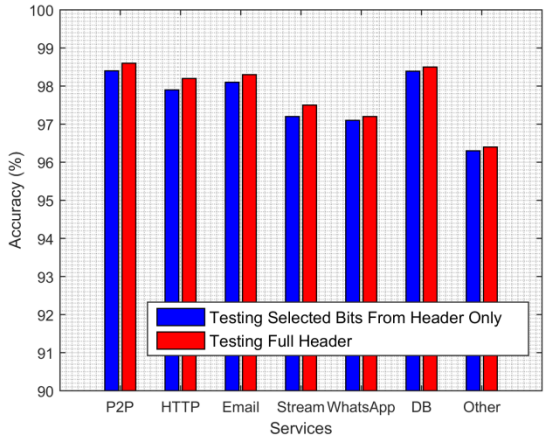

Figure. 6 The comparison of accuracy versus services by using selected and full header bit

training pattern on random step, as expected. Our approach increases the accuracy by nearby 17% if correlated with a pattern without the pretraining.

Fig. 5 presents the effect of increasing the number of sampled packets on the accuracy of the system. We tested the range from 40 to 90 sampled packets for each flow and set the steady step sampling method parameters with and without pretraining. The accuracy decreases as the number of sampled packets increases. This is because increasing the number of packets sampled improves the predictive accuracy of the statistical features. However, it is difficult for the system to learn class labels while the input dimension is more numerous due to the small training set. The accuracy with pretraining improved by nearly 10% for 25 flows, while it enhanced by around 15% for 7 flows on the accuracy without pretraining.

The improved performance metric points that it is achievable to train a suitable identifier as small as 25 flows per class when we apply our SsML. Consequently, it dramatically decreases the data set labeling and collection that require more CPU utilization, processing-time consumption, and intense action steps. To evaluate the performance and improvement of our SsML by using our novel proposal FDPHI on real-time traffic classification, we used a steady step sampling due to its high classification accuracy. Classification accuracy in our model is not affected much by using the full header or selected header bits as shown in Fig. 6 for seven grouped services.

The accuracy of the two cases (full header and selected header bits) is higher than 96%, and the difference for accuracy between them is less than0.3%. However, CPU usage, processing time, and power consumption are significantly affected by applying specific header bits more than use the full

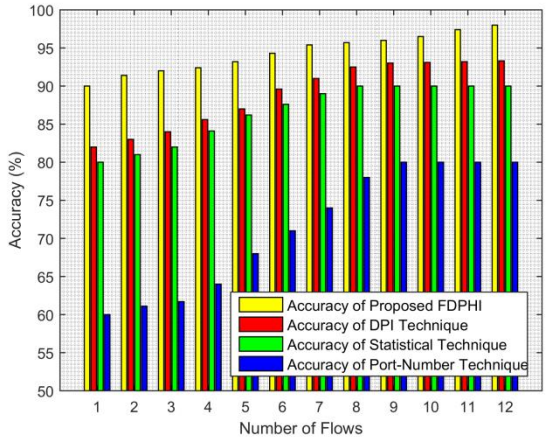

Figure. 7 Comparison of traffic classification techniques

header method.

Fig. 7 shows the comparison of the state-of-the-art traffic classification techniques. We relied on the packet header information to classify traffic flows incoming the system. As can be seen from Fig. 7, FDPHI recorded high accuracy even at one flow entering the system and increased the accuracy by increasing the number of flows. However, the traditional DPI technique obtained the second higher accuracy, and the statistical method showed a lower accuracy than the FDPHI and DPI. While the port number approach presented the lowest value of the accuracy because it depends on extracting only the port numbers from the UDP/TCP header fields of packets to classify traffic flows.

5. Performance evaluation

Our SsML classifier showed the highest performance when the steady step approach was applied. Therefore, this approach is used by the SsML to classify traffic flows based on testing full header bits and selected bits from the header. The experiment was implemented by using a computer as Fig. 8 illustrates that FDPHI needs less power to classify traffic flows. The power consumption increases as the number of flows increase that subject

Table 3. Workstation specifications used in experiment

Item	Value
CPU	Intel Xeon Bronze 3106 Processor-8Core,1.70GHz
RAM	64GB 2666MHz DDR4 LR ECC DIMM Module
Cache	11MB
OS	Microsoft Windows Server 2019
HDD	3.8TB Intel SSD S4510 Data Center SERIES 2.5IN
Power	40W (Save Mode) up to 250W (Max Mode)

a workstation in this work. This workstation has the specifications as stated in Table 3.

to classification by FDPHI. In the case of 7 flows, the method of the selected bits recorded power consumption less than the full header nearly 30%. While in the case of 25 flows, the FDPHI technique with picked header bits consumed 70% less power than the entire header. It stands to reason that SsML would require more power when the amount of data is large. Thus, CPU utilization (physically) and CPU usage (logically) rise due to the increase in implementation steps to complete the classification process.

Fig. 9 presents utilization versus the number of flows. As can be seen from the Figure, the CPU utilization increases with increasing the number of traffic flow enters the SsML classifier. At 7 flows as inputs, the CPU utilization for testing the picked header bits decreased by 14 % from the full header test. Whereas, at 25 flows with applying the FDPHI technique the CPU utilization decreased about 48% of testing full header bits. Consequently, the required energy and processing time to complete the classification task are reduced.

Fig. 10 shows applying the FDPHI technique on the SsML for testing full header bits of IPv4. The FDPHI improved the performance of the system to classify data traffic around 310% at the number of tested bits reach 160 bits. In other words, when full header bits (160 bits for one packet in IPv4) will cost processing time nearly 465 ms. While the same number by the accumulation of tested bits by the FDPHI will take approximately 175 ms, but for more than one packet. Fig. 11 describes the effect of implementing the FDPHI method on the SsML for testing full header bits of IPv6 (320 bits). The FDPHI improved the performance of the system to classify

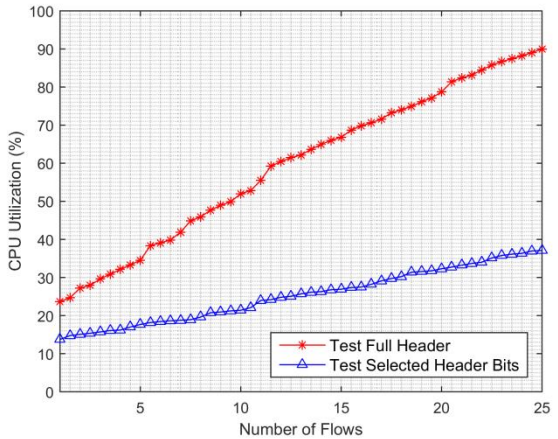

Figure. 8 Power consumption of SsML utilizes FDPHI approach when using selected and full header bits

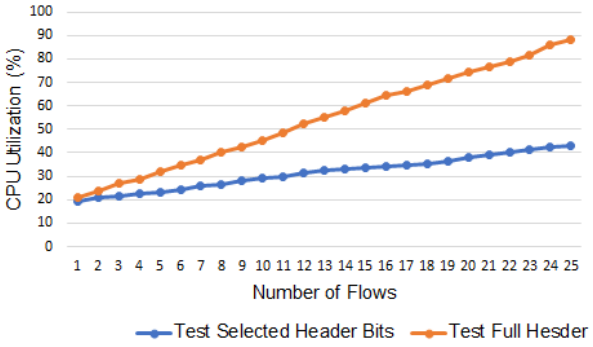

Figure. 9 CPU utilization of SsML utilizes FDPHI approach when using selected and full header bits

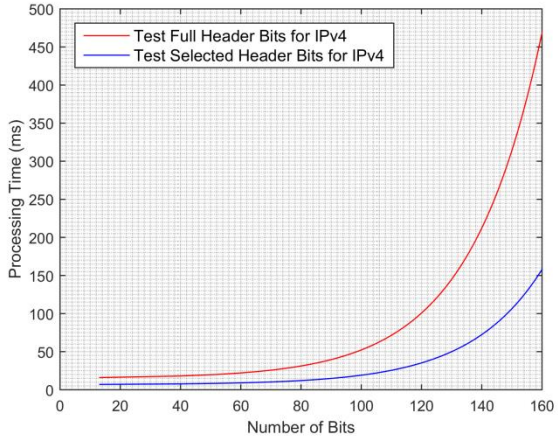

Figure. 10 IPv4 processing time with and without using FDPHI approach

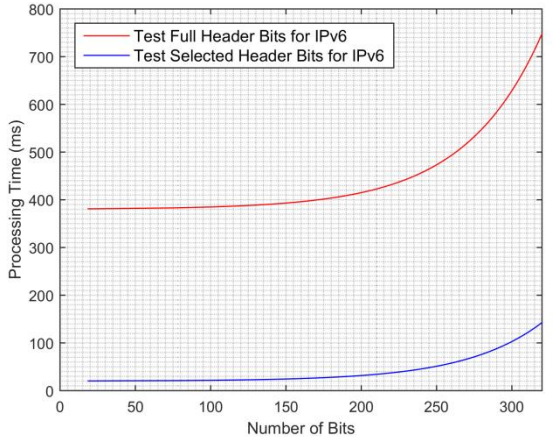

Figure. 11 IPv6 processing time with and without using FDPHI

data traffic around 595% as the number of tested bits reach 320 bits. This means full header bits will cost a processing time of almost 740 ms.

In contrast, the similar accumulated number of tested bits by the FDPHI will need around 145 ms at testing more than one packet. Therefore, the performance of the system increased. We repeated the experiments 80 times by injecting different types of the collected data.

6. Conclusion

Our novel FDPHI traffic classification approach does away with common steps, such as features selection, features extraction, and feature design which are generally utilized in the classical divide-and-conquer strategy. It employs 1D-CNN to automatically learn more representational characteristics of traffic flow types. Although current techniques of traffic classification are efficient, they still lack new creative approaches to satisfy the demands of vital issues such as real-time traffic classification, low power consumption, pace processing, etc. Our proposal suggests an unprecedented idea to classify network traffic based on selecting affected bits from the packet header, instead of testing the full header bits. The FDPHI classifies the traffic flows as well as the traditional methods, but with much less in consuming power, CPU utilization, and processing-time for both IPv4 and IPv6 header tests. The experimental results show better achievement than the most recent traffic classification methods. The FDPHI based on 1D-CNN gives very satisfactory traffic classification results in terms of energy consumption (70% less power), CPU utilization (around 48% less), and processing time (310% for IPv4 and 595% for IPv6). Based on the obtained results, we can deduce the necessity for finding an innovative design to improve and promote the traffic classification techniques. At the same time, avoid spending resources on categorizing traffic such as consumed power, processing time, and CPU usage as the FDPHI did.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, validation, formal review, research, resources, editing and Supervision have been done by 1st author, software, writing-original draft preparation, and research administration, have been done by 2nd author.

References

[1] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, “A survey on encrypted traffic classification”, In: *Proc. of International Conf. on*

Applications and Techniques in Information Securit, pp. 73–81, 2014.

[2] B. C. Park, Y. J. Won, M. S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification", In: *Proc. of NOMS 2008 IEEE Network Operations and Management Symposium*, pp. 160–167, 2008.

[3] L. Hu and L. Zhang, "Real-time internet traffic identification based on decision tree", In: *Proc. of World Automation Congress*, 2012.

[4] W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques", *In: Proc. of the 2005 ACM SIGMETRICS international Conf. on Measurement and modeling of computer systems*, pp. 50–60, 2005.

[5] P. Velan, M. Cˇ erma´k, P. Cˇ eleda, and M. Drasˇar, "A survey of methods for encrypted traffic classification and analysis", *International Journal of Network Management*, Vol. 25, No. 5, pp. 355–374, 2015.

[6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark", In: *Proc. of the 2005 Conf. on Applications, technologies, architectures, and protocols for computer communications*, pp. 229–240, 2005.

[7] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting", *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 1, pp. 5–16, 2007.

[8] F. Samie, L. Bauer and J. Henkel, "From Cloud Down to Things: An Overview of Machine Learning in Internet of Things", *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 4921-4934, 2019.

[9] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*, MIT press, 2018.

[10] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway", *IEEE Access*, Vol. 6, pp. 55380–55391, 2018.

[11] M. Latah and L. Toker, "Artificial intelligence enabled software-defined networking: a comprehensive overview", *IET Networks*, Vol. 8, No. 2, pp. 79–99, 2018.

[12] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for tcp traffic classification", *Computer Networks*, Vol. 53, No. 14, pp. 2476–2490, 2009.

[13] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "Kiss: Stochastic packet inspection classifier for udp traffic", *IEEE/ACM Transactions on Networking*, Vol. 18, No. 5, pp. 1505–1515, 2010.

[14] L. Peng, B. Yang, and Y. Chen, "Effective packet number for early stage internet traffic identification", *Neurocomputing*, vol. 156, pp. 252–267, 2015.

[15] R. Li, X. Xiao, S. Ni, H. Zheng, and S. Xia, "Byte segment neural network for network traffic classification", In: *Proc. of IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2018.

[16] A. Duque-Torres, F. Amezquita-Su´arez, O. M. Caicedo Rendon, A. Ord´o˜nez, and W. Y. Campo, "An approach based on knowledge defined networking for identifying heavy-hitter flows in data center networks", *Applied Sciences*, Vol. 9, No. 22, pp. 48084827, 2019.

[17] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions", *ACM SIGCOMM Computer Communication Review*, Vol. 35, No. 4, pp. 217–228, 2005.

[18] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected means of protocol inference", In: *Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement*, pp. 313–326, 2006.

[19] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning", *Soft Computing*, Vol. 24, No. 3, pp. 1999–2012, 2020.

[20] K. Chokkanathan and S. Koteeswaran, "An integrated approach for network traffic analysis using unsupervised clustering and supervised classification", *International Journal of Internet Technology and Secured Transactions*, Vol. 9, No. 4, pp. 517–536, 2019.

[21] M. A. Lopez, D. M. Mattos, O. C. M. Duarte, and G. Pujolle, "A fast unsupervised preprocessing method for network monitoring", *Annals of Telecommunications*, Vol. 74, No. 3-4, pp. 139–155, 2019.

[22] S. A. Kokatnoor and B. Krishnan, "Self-supervised learning based anomaly detection in online social media", *International Journal of Intelligent Engineering and Systems*, Vol.13, No.3, pp. 446-456, 2020.

[23] J. Zhang, C. Chen, Y. Xiang, and W. Zhou, "Robust network traffic identification with unknown applications", In*: Proc. of the 8th ACM SIGSAC symposium on Information,*

Computer and Communications Security, pp. 405–414, 2013.

[24] A. S. Iliyasu and H. Deng, "Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks", *IEEE Access*, Vol. 8, pp. 118–126, 2019.

[25] Li, G. Li, and X. Yu, "A fast traffic classification method based on sdn network", In: *Proc. of 4th Int. Conf. Electron., Commun. Netw*., pp. 223–229, 2015.

[26] Rezaei, Shahbaz, and Xin Liu. "Deep learning for encrypted traffic classification: An overview", *IEEE Communications Magazine*, Vol. 57, No.5, pp. 76-81, 2019.

[27] P. Maniriho, L. J. Mahoro, E. Niyigaba, Z. Bizimana, and T. Ahmad, "Detecting intrusions in computer network traffic with machine learning approaches", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp.433-445, 2020.

[28] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things", *IEEE Access*, Vol. 5, pp. 18 042–18 050, 2017.

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans", *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

[30] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks", In: *Proc. of IEEE International Conf. on Intelligence and Security Informatics (ISI)*, pp. 43–48, 2017.